

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (Original) A method of managing memory, comprising:

maintaining a pool of memory addresses for writing data to a memory;

retrieving memory addresses from the pool in response to write requests from data sources;

replenishing the memory addresses in the pool when the pool is emptied; and

incrementing a first counter value in response to each replenishment of the memory addresses in the pool.
2. (Original) The method of claim 1, further comprising:

determining whether a reset has been initiated; and

replenishing the memory addresses in the pool and resetting the first counter value based on the determination.
3. (Currently amended) The method of claim 1, further comprising:

storing the retrieved memory addresses in a buffer;

writing data to the memory addresses stored in the buffer; and

incrementing a second counter value when the data has been written to all the addresses stored in the buffer.

4. (Original) The method of claim 3, further comprising:

passing the first and second counter values to the data sources in response to each of the write requests.

5. (Original) The method of claim 4, further comprising:

receiving read requests from the data sources, each of the read requests comprising the first and second counter values; and

returning error messages to the data sources, indicating that the data written to the memory addresses has been aged, based on the read request first and second counter values.

6. (Original) A network device, comprising:

a memory configured to:

store a pool of memory addresses for writing data to the memory, and

store a first counter value; and

a memory allocation unit configured to:

retrieve memory addresses from the pool in response to write requests from data sources,

replenish the memory addresses in the pool when the pool is emptied, and

increment the first counter value in response to each replenishment of the memory addresses in the pool.

7. (Original) The network device of claim 6, the memory allocation unit further configured to:

determine whether a reset has been initiated; and
replenish the memory addresses in the pool and reset the first counter value based on the determination.

8. (Currently amended) The network device of claim 6, the memory further configured to:

maintain a buffer, and
store a second counter value; and
the memory allocation unit further configured to:
store the retrieved memory addresses in the buffer,
write data to the memory addresses stored in the buffer, and
increment the second counter value when the data has been written to all the addresses stored in the buffer.

9. (Original) The network device of claim 8, the memory allocation unit further configured to:

pass the first and second counter values to the data sources in response to each of the write requests.

10. (Original) The network device of claim 9, the memory allocation unit further

configured to:

receive read requests from the data sources, each of the read requests comprising the first and second counter values; and

return error messages to the data sources, indicating that the data written to the memory addresses has been aged, based on the read request first and second counter values.

11. (Currently amended) A method of managing memory, comprising:

maintaining a pool of memory addresses for writing data to a memory;

retrieving memory addresses from the pool and storing the retrieved memory addresses in a buffer;

writing the data to memory addresses stored in the buffer; and

incrementing a first counter value when the data has been written to all the addresses stored in the buffer.

12. (Original) The method of claim 11, wherein the buffer comprises a circular buffer.

13. (Original) The method of claim 11, wherein the pool comprises a first-in-first-out queue.

14. (Original) The method of claim 11, further comprising:

receiving read requests comprising tag values.

15. (Currently amended) The method of claim 14, further comprising:
reading selected ones of the data from one or more selected memory addresses if the tag values match other tag values stored in a table.
16. (Original) The method of claim 15, further comprising:
deallocating the one or more selected memory addresses.
17. (Original) The method of claim 14, further comprising:
returning an error message indicating that the data from the one or more selected memory addresses has been aged when the tag values fail to match a tag value stored in a table.
18. (Currently amended) A network device, comprising:
a memory configured to:
store a pool of memory addresses for writing data to the memory,
store a first counter value, and
maintain a buffer; and
a memory allocation unit configured to:
retrieve memory addresses from the pool and store the retrieved memory addresses in the buffer,
write data to memory addresses stored in the buffer; and
increment the first counter value when the data has been written to all the

addresses stored in the buffer.

19. (Original) The network device of claim 18, wherein the buffer comprises a circular buffer.

20. (Original) The network device of claim 18, wherein the pool comprises a first-in-first-out queue.

21. (Original) The network device of claim 18, the memory allocation unit further configured to:

receive read requests comprising tag values.

22. (Currently amended) The network device of claim 21, the memory allocation unit further configured to:

read selected ones of the data from one or more selected memory addresses if the tag values match other tag values stored in a table.

23. (Original) The network device of claim 22, the memory allocation unit further configured to:

deallocate the one or more selected memory addresses.

24. (Currently amended) ~~A data structure encoded on a computer-readable medium~~

system, comprising:

a data structure encoded on a computer-readable medium, comprising:

first data comprising a pool of addresses for at least one of writing and reading
other data to and from a memory₁[[;]] and

second data comprising a counter value that indicates a number of times the
pool has been emptied of memory addresses and then replenished; and

a unit configured to use the first and second data in allocating or de-allocating
memory for memory read/write operations.

25. (Currently amended) A system ~~data structure encoded on a computer-readable
medium~~, comprising:

a data structure encoded on a computer-readable medium, comprising:

first data comprising a list of memory addresses obtained from a memory
address pool₁[[;]] and

second data comprising a counter value that indicates a number of times other
data has been written to all the memory addresses in the list; and

a unit configured to use the first and second data in allocating or de-allocating
memory for memory read/write operations.

26. (Currently amended) The system ~~data structure~~ of claim 25, the data structure further
comprising:

third data comprising pointers to each of the memory addresses in the list,

wherein the unit is further configured to use the third data in allocating or de-allocating memory for memory read/write operations.

27. (Currently amended) The system data structure of claim 25, the data structure further comprising:

third data comprising a counter value that indicates a number of times the memory address pool has been emptied of memory addresses and then replenished,

wherein the unit is further configured to use the third data in allocating or de-allocating memory for memory read/write operations.

28. (Currently amended) The system data structure of claim 25, the data structure further comprising:

third data comprising flags that indicate whether each of the memory addresses have been allocated for a memory write operation,

wherein the unit is further configured to use the third data in allocating or de-allocating memory for memory read/write operations.

29. (Original) A system for managing memory, comprising:

means for maintaining a pool of memory addresses for writing data to a memory;

means for retrieving memory addresses from the pool in response to write requests from data sources;

means for replenishing the memory addresses in the pool when the pool is emptied;

and

means for incrementing a first counter value in response to each replenishment of the memory addresses in the pool.